

UNISYS

**A Series
Communications
Management System
(COMS)
Capabilities
Manual**

Release 3.9.0

September 1991

Priced Item

U S America
8600 0627-000

UNISYS

A Series

**Communications
Management System
(COMS)**

**Capabilities
Manual**

Release 3.9.0

September 1991

Priced Item

U S America
8600 0627-000



UNISYS

**A Series
Communications
Management System
(COMS)
Capabilities
Manual**

Copyright © 1991 Unisys Corporation

All rights reserved.

Unisys is a registered trademark of Unisys Corporation.

Release 3.9.0

September 1991

Priced Item

U S America
8600 0627-000

The names, places, and/or events used in this publication are not intended to correspond to any individual, group, or association existing, living, or otherwise. Any similarity or likeness of the names, places, and/or events with the names of any individual, living or otherwise, or that of any group or association is purely coincidental and unintentional.

NO WARRANTIES OF ANY NATURE ARE EXTENDED BY THE DOCUMENT. Any product and related material disclosed herein are only furnished pursuant and subject to the terms and conditions of a duly executed Program Product License or Agreement to purchase or lease equipment. The only warranties made by Unisys, if any, with respect to the products described in this document are set forth in such License or Agreement. Unisys cannot accept any financial or other responsibility that may be the result of your use of the information in this document or software material, including direct, indirect, special or consequential damages.

You should be very careful to ensure that the use of this information and/or software material complies with the laws, rules, and regulations of the jurisdictions with respect to which it is used.

The information contained herein is subject to change without notice. Revisions may be issued to advise of such changes and/or additions.

Correspondence regarding this publication may be forwarded using the Product Information card at the back of the manual, or may be addressed directly to Unisys, Product Information, 25725 Jeronimo Road, Mission Viejo, CA 92691.

Section 1

Overview of COMS

The Communications Management System (COMS) is a message control system (MCS) and an application controller that supports communications and transaction processing for a terminal network on the A Series systems.

It is a member of the InterPro™ (Interactive Productivity) Series of software. COMS supports the processing of multiprogram transactions as well as single-station and multistation remote files. The advanced modular design of COMS makes it a flexible and highly efficient system that handles high-volume transaction throughput.

There are two versions of COMS: COMS (Full-Featured) and COMS (Kernel). COMS (Kernel) is a transitional version of COMS that permits only minimum features. All the benefits and features covered in this manual can be found in the full-featured COMS product, hereafter referred to as COMS.

COMS (Kernel) does not support the following features:

Feature	Description
Easy user interface	The ability to define and maintain the COMS configuration file. This file contains the characteristics of the COMS environment. For COMS (Kernel), this environment is fixed.
integrated programmatic interface	Because of the inability to declare direct windows that use transaction processing application programs, COMS input and output headers cannot be used for application programs with COMS (Kernel).
Routing flexibility	COMS (Kernel) cannot be customized to use Transaction Based Routing.
Database recovery	COMS (Kernel) does not maintain recovery information for databases and cannot resubmit transactions to any database.
Efficiency monitoring	COMS (Kernel) has no statistics capabilities.
Migration aids	Because the COMS transaction control language (TCL) formatter uses customized COMS configurations, it cannot be used with COMS (Kernel).

Migrating from COMS (Kernel) to COMS requires no changes to the COMS environment; it simply expands your capabilities.

InterPro is a trademark of Unisys Corporation.

Section 2

COMS Features

COMS is an integrated, multifaceted product that uses current software technology to efficiently perform functions such as message routing, application control, transaction processing, and terminal network control.

The features that are provided by COMS are covered in this section, categorized by the following functional areas:

- User interface
- Transaction processing
- Communications
- Database management
- Security

User Interface Features

COMS provides the following user interface features:

Feature	Description
Multiple environments	<p>COMS windows enable you to run multiple program environments simultaneously from a single station. For example, an application that functions in another MCS environment, an application developed as a remote-file program, and an application developed for the COMS environment can be operating at the same time. You can view and communicate with one environment at a time while the others continue to run.</p> <p>An access to a program environment within a window is called a <i>dialog</i> or <i>window dialog</i>. COMS enables you to have up to eight dialogs within any given window. Each dialog constitutes an independent session with a program environment. The combination of windows and window dialogs enables you to access many program environments from a single station.</p>

continued

COMS Features

continued

Feature	Description
Predefined windows	<p>The Menu-Assisted Resource Control (MARC) window enables you to operate your system by using a menu-driven interface. The MARC window is always available in COMS.</p> <p>The COMS Utility window enables you to define and modify the specifications of the COMS configuration file.</p> <p>The Command and Edit (CANDE) window enables you to use CANDE, if CANDE is installed on your system.</p> <p>The Generalized Message Control System (GEMCOS) window enables you to use GEMCOS, if GEMCOS is installed on your system.</p>
Types of programs and windows	<p>Transaction processing programs are the most flexible and efficient applications COMS controls. These programs are contained in direct windows. Transaction processing programs and direct windows are predefined by COMS, or are defined by you through the COMS Utility. The MARC window and an accounts payable application developed to use COMS input and output headers are examples of this type of window and program.</p> <p>Other MCS programs on your system are compatible with COMS. COMS enables MCS programs to be accessed through MCS windows. MCS windows are predefined by COMS, or are defined by you through the COMS Utility. The CANDE window and your own MCS window are examples of MCS windows.</p> <p>Remote-file programs allow access to COMS from standard logical I/O interfaces. Remote-file programs are either controlled by COMS through remote-file windows, or they are initiated by another source. For example, remote-file programs can be initiated as a result of a MARC or CANDE <i>RUN</i> command or a Work Flow Language (WFL) <i>RUN</i> statement. In the latter case, COMS dynamically creates a window to allow access to these remote files.</p>
User-defined windows	<p>As a COMS user, you can develop your own program environments. Developing these environments includes defining the characteristics of the window in the COMS configuration file and writing one or more programs to handle the processing of messages.</p>
Window close action	<p>This feature enables you to define the action that is taken when a user is currently on a window and the window closes. You can specify the close action by station or by usercode. You can choose any of the following close action options: close to a specified window, close to a previous window, or close and wait for the closed window to open again.</p>

continued

*continued***Feature****Description**

Easy configuration

The COMS Utility enables you to establish or reconfigure the COMS environment without bringing down the system. The requested changes are made to the COMS configuration file, a central repository of information about the COMS environment. The COMS Utility enables you to enter the changes by using menus and screens or by entering commands.

The COMS Utility gives you the flexibility to add special processing functions that can be applied at any point during transaction processing. Because this utility is dynamic, you can easily use it to test new or modified modules by routing messages from selected stations to a destination before integrating the modules into the system.

You can limit the use of the COMS Utility to a specific person, a group of people, a station, or a group of stations, or any combination of these restrictions.

Figure 2-1 shows the Window Activity screen, in which you can define each window in the COMS environment.

Control over COMS operations

An authorized user can perform a variety of functions by using the online COMS commands. The user can initiate these commands by selecting the commands from a MARC window screen, or by entering the commands from any window accessible to the user in the COMS environment.

Any authorized user can control the following tasks:

- Starting or ending a session
- Sending messages to other stations or to the operator display terminal (ODT)
- Moving from one window dialog to another
- Entering text for another window dialog
- Determining which windows are available to the user
- Determining which programs are running in a specific window

In addition, a control-capable operator or a user at a control-capable station can use COMS commands for the following tasks:

- Controlling the COMS network
- Controlling transaction trails
- Retrieving information about the COMS environment

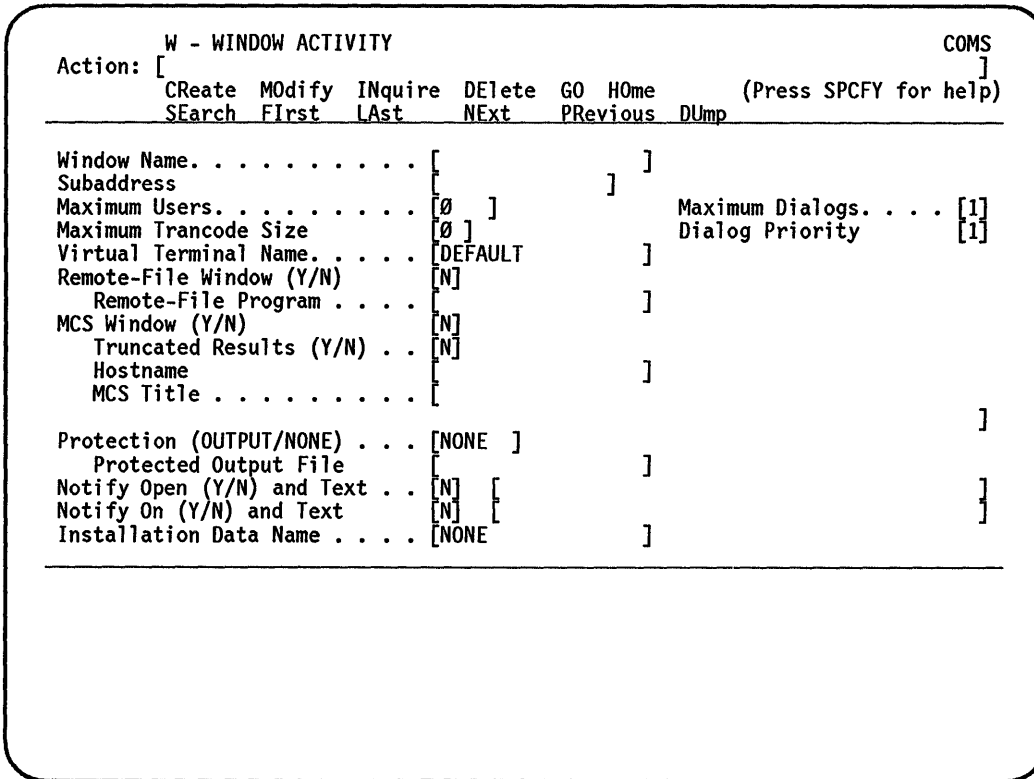


Figure 2-1. Window Activity Screen

Transaction Processing Features

COMS provides the following transaction processing features when you are using transaction-processing programs and direct windows:

Feature	Description
Integrated programmatic interface	COMS input and output headers provide a simple interface with application programs written in ALGOL, COBOL74, Pascal, and RPG.
High efficiency	COMS enables you to develop transaction-processing systems that have low overhead, efficient processing, and high throughput rates by integrating portions of COMS with the operating system.
Dynamic program control	COMS enables you to control application programs within a direct window without bringing down the COMS system. You can control the number of copies of the program that can be running at one time, the time that a copy of the program is to be executed or terminated, and the number of words that can be in the input queue before additional words are tanked (stored temporarily) to disk.

continued

continued

Feature	Description
Transaction routing	COMS enables you to develop applications that determine the destination of the message and any special processing based on the transaction code (trancode) present in the message. The routing of messages is based on the agenda associated with each trancode. Additionally, you can control the usercodes and stations that can use a given trancode.
Efficiency monitoring	The COMS Statistics program enables you to monitor the efficiency of your COMS environment. You can either receive reports online on the Statistics window or print out the reports. The available information includes the following: <ul data-bbox="846 604 1450 823" style="list-style-type: none"><li data-bbox="846 604 1450 663">• The amount of processor time that is consumed by each copy of a program<li data-bbox="846 674 1450 732">• The length of time a copy of a program has been in the mix<li data-bbox="846 743 1450 779">• The number of messages waiting for a program<li data-bbox="846 789 1450 823">• The average response time for a program
Routing flexibility	The modular design of COMS makes it flexible and highly efficient. This flexibility enables you to select features and add special processing for the needs of your site. <p data-bbox="846 947 1450 1142">COMS enables you to route messages by using an agenda. Any agenda can have independently defined pieces of code, called processing items, associated with it. With COMS, you have the additional flexibility of being able to use an agenda and its processing items before a message reaches a direct-window program (preprocessing) and after the message leaves a direct-window program (postprocessing).</p> <p data-bbox="846 1163 1450 1243">The combination of processing items and agendas enables you to develop a customized COMS environment that has flexibility both now and in the future.</p>
Migrating from GEMCOS to COMS	The COMS transaction control language (TCL) formatter processing item enables you to use GEMCOS message formatting and paging in the COMS environment.

Communications Features

COMS provides the following communications features:

Feature	Description
CP 2000 interface	<p>COMS Terminal Transport Protocol (TTP) Protocol-Specific Handler (PSH) enables you to use CP 2000 data communications processors, which enable you to easily connect terminals to multiple large-system hosts. Thus, if one large-system host goes down, you can switch quickly to another running large-system host.</p> <p>Another advantage of the CP 2000 is that it assumes tasks normally handled by the mainframe, such as routing and station error processing. As a result, the throughput of the mainframe is improved.</p> <p>The interface that is used is the Terminal Transport Protocol (TTP). COMS supports both the TTP3 and TTP5 versions of the TTP.</p>
ANSI X3.64 terminal devices	<p>COMS fully supports ANSI X3.64 devices connected to a CP 2000 interface. Full support includes the capability of displaying Screen Design Facility Plus (SDF Plus) screens.</p>
Automatic dial-out capability	<p>This feature enables an application developed for the COMS environment to connect a station to the system programmatically. With this capability you can help control remote-station access to the COMS environment.</p>

Database Management Features

COMS provides the following database management feature:

Feature	Description
Database recovery	<p>If you are updating a Data Management System II (DMSII) or Semantic Information Manager (SIM) database, you can let COMS synchronized recovery manage a transaction trail that is automatically synchronized with the DMSII or SIM audit trail when a transaction-state termination, system crash, or rollback has occurred. During recovery, COMS reapplies the transactions to the database in the same order in which they were originally submitted and reinstates all messages that have been saved by the protected input queue feature. While recovery of a database is in progress, all programs within the COMS environment that do not access that database can proceed normally.</p> <p>To support recovery, direct-window programs must include specific recovery routines.</p>

Security Features

COMS provides the following security feature:

Feature	Description
Security	COMS enables you to restrict access to the COMS Utility, to the control-capable COMS commands, to COMS windows, and to the stations. In addition, COMS enables you to restrict access to a given trancode.

Section 3

COMS Architecture

The concept of libraries was used to develop modular processing for COMS. The COMS libraries contain a set of procedures that can be accessed individually by other programs. COMS relegates mainstream message-control activities to a series of hierarchical libraries, thereby lowering the amount of process initiation. Most COMS functions are performed within library procedures that run on top of the stack of the calling program or on top of data communications controller (DCC) stacks. Low-volume auxiliary functions are performed in active COMS stacks. This lowering of process initiation contributes to the high efficiency of COMS.

The basic architectural design elements of COMS follow:

Elements	Description
COMS library	This library contains the COMS object code, as well as service-function procedures that enable an application program to obtain information from the configuration file. The COMS library also supports changes to the configuration-file tables while COMS is running. Furthermore, this library contains the dynamic selection procedures for linking callers to the router library, the transaction processor (TP) library, a database (DB) library, or the agenda processor library.
COMS configuration file	The COMS configuration file contains descriptions that define all the entities related to COMS, such as usercodes, stations, windows, programs, agendas, and processing items. When COMS is installed, tables are built for the predefined entities and elements. After those tables are built, you can define and maintain your own application in the configuration file by using the COMS Utility.
Router library	This internal COMS library contains procedures called by the operating system. The input-router procedure is used by the DCC stack for processing all input-message and output-message results associated with any COMS-controlled stations. The output-router procedure is used by logical I/O when handling output from remote-file programs.
COMS control and database control	The COMS control program and database (DB) control program perform auxiliary asynchronous functions. Every defined database has a separate DB control that ensures a reliable, active stack is available when application programs access the database. COMS control ensures the same stack availability for application programs that do not access a database. Additionally, COMS control and DB control monitor the state of COMS, such as the running or not-running state. Finally, COMS control runs and controls application programs, and controls slow-routing functions, such as truncate errors, security errors, data-communications errors, and logging.

continued

continued

Elements	Description
Database libraries and transaction-processor library	<p>A database library provides two basic functions: support for the user program interface by using the data communications interface (DCI) library interface mechanism, and support for recovery of a DMSII or SIM database.</p> <p>Programs that use the DCI library but do not require the database functions provided by the DB library maintain the same relationship with the TP library as database programs maintain with their DB library. Program control and queue implementation are the same for all application programs.</p>
COMS input/output handler	<p>COMS minimizes disk I/O operations, which improves performance. When an operator requests that data from a specific window be stopped, COMS writes the output messages to disk (this process is called tanking). When the window is allowed to send data again, COMS reads the messages from disk and sends them to the appropriate station. The COMS I/O handler handles the I/O completions on behalf of COMS control, DB control, user application programs, and the router library.</p>
Agenda processor library	<p>This library calls procedures in processing-item libraries. It is called from the DB libraries, the TP library, or COMS control. The agenda processor library executes the processing items that pertain to given agendas. Dynamic library selection from the COMS library is used to link the DB libraries, the TP library, and COMS control to the agenda processor library.</p>
Database transaction trails and COMS transaction trail	<p>A transaction trail for a database is a file, generated by a DB library, that contains a series of time-ordered transactions that can be reapplied to the database. COMS reapplies the transactions to provide synchronized recovery of the database in the event of a transaction state termination, system crash, or rollback. The file optionally provides a journal of query transactions associated with the database and provides statistical information on a transaction-by-transaction basis, which you can use for security and accounting purposes. Each DB library generates its own transaction trail.</p> <p>A COMS transaction trail is a file, generated by the TP library, that reflects such information as beginnings of jobs and ends of jobs. The file optionally provides a journal of query transactions not associated with any database and provides statistical information on a transaction-by-transaction basis, which you can use for security and accounting purposes.</p>
Protocol-specific handler	<p>The protocol-specific handler (PSH) library performs all terminal-connection control independent of COMS. This eliminates the need by COMS to know specific protocol details and allows for flexible support of different terminal connection paths by COMS.</p>

Section 4

Hardware and Software Requirements for COMS

This section discusses the hardware and software that is needed to run COMS and some of the information resources that are available.

Prerequisite Hardware

To run COMS you need the following hardware:

- An A Series system
- A disk drive
- 3.96 megabytes of disk storage for COMS files

To use the Utility window to reconfigure COMS, you need a forms-capable terminal device equivalent to a T27 terminal.

The amount of memory COMS uses varies depending on how you use and configure the network. As an example of representative memory use, if you define a network consisting of 45 stations, 20 windows, and 28 active users, COMS uses approximately 9,000 words of save memory. If you define the maximum number of elements the configuration file allows, COMS uses approximately 17,000 words of save memory plus 3,000 words of save memory for each open database. (Refer to the *A Series Memory Subsystem Administration and Operations Guide* for information on save memory.)

Prerequisite Software

To run COMS you need the following A Series software:

- A Series systems COMS (Kernel) or COMS (Full-Featured) software.
- A Series systems Menu-Assisted Resource Control (MARC) software. COMS (Kernel) is provided with the MARC software.

The following software is necessary to compile COMS:

- A Series systems SYSTEM/DMALGOL software
- A Series systems SYSTEM/BINDER software
- A Series systems SYSTEM/TRINTERFACE software

COMS Publications

The following publications describe the various aspects of COMS in detail, including one book that is a composite document index to all the COMS guides:

- *A Series Communications Management System (COMS) Configuration Guide* (form 8600 0312)
- *A Series Communication Management System (COMS) Documents Index* (form 8600 0635)
- *A Series Communications Management System (COMS) Migration Guide* (form 8600 1567)
- *A Series Communications Management System (COMS) Operations Guide* (form 8600 0833)
- *A Series Communications Management System (COMS) Programming Guide* (form 8600 0650)

The following references might also be useful:

- *A Series ALGOL Programming Reference Manual, Volume 2: Product Interfaces* (form 8600 0734)
- *A Series COBOL ANSI-74 Programming Reference Manual, Volume 2: Product Interfaces* (form 8600 0130)
- *A Series Memory Subsystem Administration and Operations Guide* (form 1169836)
- *A Series Pascal Programming Reference Manual, Volume 2: Product Interfaces* (form 8600 1294)
- *A Series Report Program Generator (RPG) Programming Reference Manual, Volume 2: Product Interfaces* (form 8600 0742)

Summary

The use of modules, front-end processors, agendas, libraries, windows, menus, screens, a utility for online system definition and maintenance, and a statistics-gathering mechanism contribute largely to making COMS an easy-to-use system that is flexible and highly efficient. COMS is designed to satisfy both present and future MCS needs.





86000627-000